Diese Website verwendet Cookies von Google, um Dienste anzubieten und den Traffic zu analysieren. Deine IP-Adresse und dein User-Agent werden zusammen mit Messwerten zur Leistung und Sicherheit für Google freigegeben. So können Nutzungsstatistiken generiert, Missbrauchsfälle erkannt und behoben und die Qualität des Dienstes gewährleistet werden.

WEITERE INFORMATIONEN OK

Monday, June 11, 2007

Nanobarter

There have been over the years several plans and attempts to develop very fined grained markets online. There are several barriers to the success of such markets. An important barrier recently raised by Zooko in his comments on the Tahoe peer-to-peer disk backup project is the vulnerability of and to a centralized mint issuing money.

One possible answer to central mint vulnerability is bit gold -- a currency the value of which does not depend on any particular trusted third party. Another alternative is an object barter economy.

The key ideas of this nanobarter scheme are

(1) the stuff to be traded (in Tahoe, disk space for backup) is represented by digital barter certificates (same protocol as digital cash, but every node is a "mint" that issues its own barter notes), and

(2) default barter order and an agent ("market translator") that translates user behavior into barter orders. In the disk space economy, the default barter order might be to simply do a periodic barter that backs up N gigabytes of other peoples' disks in exchange for N gigabytes of their own. Many more sophisticated barter orders are also possible.

If the reader is familiar with Menger's account of the origin of money from barter, this scheme is quite in the spirit of his scenario -- except that we reduce the transaction costs of barter by brute force automation instead of by making everybody choose a single currency.

The transaction log and accounts are presented to the user in terms of a "pet currency"; the market translator automatically converts all different kinds of barter note prices into their pet currency values whenever prices need to be presented to the user.

Every computer on the network (called a "node") runs a "mint" that issues "currency" (barter notes) backed by its commodity (e.g. disk space). In a simple system all disk space barter notes are treated as equivalent. Or there might be L different currencies corresponding the the L different kinds of leases in Tahoe. (In Tahoe a certain amount of disk space on a foreign disk is "leased" for a certain period of time). Indeed, a barter note is simply a lease in bearer form -- it can mean "I promise to bearer to provide G gigabytes of disk space for D days", or whatever the terms of the lease is.

In a simple system, the barter note may simply be a ticket that never gets traded, merely issued and used. In a slightly more advanced

About Me

Nick Szabo

"A premier thinker about history, law and economics, and the lessons they have for security." -- Adam Shostack, Emergent Chaos

"Szabo comes out with these essays that leave me in awe." -- Brian Dunbar

"Reading material that is eclectic, challenging and endlessly fascinating." -- Sean McGrath, Propylon

"Like most blogs worth my attention, this blog is updated only infrequently. That is because the authors of blogs worth my attention only post when they have something to say that is true, relevant and not already known by their audience. Most of the human race does not have the skill to know when an idea has these three properties. The skill is particularly rare in the fields of politics and economics, which is why this blog is such a rare and valuable thing." -- Richard Hollerith

View my complete profile

system they trade but only at par value. A gigabyte is a gigabyte regardless of whose server its on -- this is a very simple proxy measure that excludes service quality from automated consideration. Since this is a nanomarket, there is normally no opportunity for the user to intervene with a more sophisticated or subjective judgment. Even a crude proxy measure, if full automated, may be sufficient for a nanomarket to outperform the non-nanomarket status quo (no transactions at all or the use of resource allocation algorithms, although the latter in a broad sense can be considered to be competing nanobarter systems).

In a more sophisticated system (probably overkill for the purposes of Tahoe) some disk space notes trade at a discount because their backup services are unreliable. Bots "ping" the backup services provided by nodes to gather statistics on their reliability, and then buy reliable and sell unreliable notes. There are O((LN)^2) automated currency exchange products which these bots trade. The mental transaction cost problem caused by having O((LN)^2) prices with LN currencies is thus solved underneath the covers by these automated trading bots. The resulting trades are presented to users, if necessary, in terms of pet currencies, and we can have a large barter economy without the mental overhead of all those prices.

To avoid the transaction costs of thinly traded markets, the bots might come to prefer the notes of one or a few services as "intermediate commodities" asMenger described, and most of the markets might become unused, leading to O(LN) actively traded markets -- an economy with a constant number of currencies and LN prices. But that's an entirely optional process that can be allowed to emerge. And with the right reliability-ping and arbitrage bots I suspect the transaction costs of thinly traded markets might be quite small, so that there is no compelling reasoning for a centralized currency to emerge and the added reliability of multiple currencies can be retained without the hassle (mental transaction costs) of users having to deal with multiple currencies.

There are few computational transaction cost barriers left to developing nanotransactions -- the biggest is network delay time. The largest remaining barrier to nanomarkets is, for most kinds of transactions, mental transaction costs. User audits of nanotransactions cannot be both frequent and at fine granularity, or mental transaction costs quickly come to dwarf the value added by the market. Any problems with nanomarkets that might require such audits must be handled in a highly automated fashion.

The approaches to designing this automation all seem to start with developing reasonable proxy measures of service value. For nanomarkets it is far more important that these be measurable in fully automated fashion than that they be terribly accurate. After good proxy measures have been developed, one must obtain or estimate user preferences in terms of these measures. Obtaining preferences directly from the user has to be done in with traditional value granularites, otherwise mental transaction costs dominate. Alternatively, further proxy measures can be made that estimate user economic preferences from their normal input behavior. These are then compiled into automated nanotransactions with the market translator.

Posted by Nick Szabo at 8:45 PM

14 comments:

Anonymous said...

That sound you hear is history repeating itself...

In a valiant attempt to circle this whole system back to its MojoNation roots it seems a lot of the wrong lessons are being learned and repeated. Zooko is mostly incorrect in stating that a centralized mint left the system untrustworthy and unable to survive. As designed, if the central bank disappeared the nodes would continue to trade but would just increase credit limits (e.g. continue to trade IOUs and favor peers in a manner that would attempt to clear outstanding balances.) As far as the issue of user trust in a

Blog Archive

- ► 2018 (1)
- ► 2017 (3)
- ► 2016 (4)
- ► 2015 (3)
- ► 2014 (3)
- ► 2013 (3)
- ► 2012 (8)
- ► 2011 (12)
- ► 2010 (9)
- ► 2009 (29)
- ► 2008 (55)
- ▼ 2007 (47)
  - ► December (1)
  - ► October (2)
  - ► September (8)
  - ► August (10)
  - ► July (4)
  - ▼ June (4)
    - A very brief history of due process
    - Nanobarter
    - In defense of the judicial branch
    - Why Justices can be rude
  - ► May (8)
  - ► April (1)
  - ► March (2)
  - ► February (4)
  - ► January (3)
- ► 2006 (130)
- ► 2005 (44)

centralized mint goes, the simple truth is that the only way such a system will grow to the point where there are enough agents invested in the system to make it self-sustaining is if someone steps up and provides the bulk resources necessary to make their currency the standard interchange between untrusted peers.

A centralized mint was not what we set out to create, it was simply a practical necessity for getting something out the door. Alternatives are far too complex for the marginal benefit they provide and when attempting to design around this reality you end up with something as complex as what has been proposed here.

The smartest thing Bram did when stripping down MojoNation to create BitTorrent was conforming the digital resource mechanism to the actual behavior of the users. In MN we tried to use the resource market to incentivize users to do what we wanted (stay connected and provide resources) instead of using it as a tool the shape internal behavior for an already existing desire (e.g. people who wanted to pirate music, software, and movies.) For BitTorrent the cost of this simplification is non-transferability of accumulated credit, but this is a small price to pay if the end result is a much simpler design and actually getting something shipped...

9:21 PM

nick said...

"the only way such a system will grow to the point where there are enough agents invested in the system to make it self-sustaining is if someone steps up and provides the bulk resources necessary to make their currency the standard interchange between untrusted peers."

This is only true if the transaction costs of barter turn out to be too high. Menger and others (and I, here) have explained why transaction costs of in-kind transactions are so high offline. This is what makes money so valuable offline. As I argued in the blog post, this may well not be true for largely fungible online services like disk space. The transaction costs of barter may be so low that adding money isn't much of a win. And if the legal system actually considers the money to be money, it comes with a very costly legal overhead. See the e-gold problems that Ian Grigg has documented well.

"Alternatives [to a centralized mint] are far too complex for the marginal benefit they provide..."

Replicating the mint to every node doesn't itself much add to the overall code complexity. Adding barter note exchange markets, and automating the whole thing so that the mints and exchanges are largely invisible to the user, might. But AFAIK nothing like this has ever been tried.

"The smartest thing Bram did when stripping down MojoNation to create BitTorrent was conforming the digital resource mechanism to the actual behavior of the users."

This conformation is what a market does quite well when it properly reflects preferences of users, but doesn't when it doesn't. This brilliance of BitTorrent conists in creating a very simple but fully automated proxy measure -- i.e. a crude but effective approximation of of user preferences that doesn't require specific input from the user. This allowed barter to be fully automated, even if the efficiency of the resource allocation falls very far below the theoretical market optimum.

But it may also be brilliant to add to the under-the-covers complexity in order to increase the efficiency and flexibility of the overall system -- to make it more like a market.

6:26 PM

Anonymous said...

"The transaction costs of barter may be so low that adding money isn't much of a win. And if the legal system actually considers the money to be money, it comes with a very costly legal overhead. See the e-gold problems that Ian Grigg has documented well."

Avoiding this issue is why we never tried to tie the credits in MojoNation to any sort of "real" currency; you never _ever_ want to open this Pandora's box if you can help it. The transaction costs of in-kind digital resource transactions are actually quite low, as MojoNation revealed back in the day, but the settlement costs are not as easy. We had the advantage of being able to perform this transaction by converting to the central currency, but you want to throw this out and introduce additional complexity for a perceived, but not proven, benefit.

"Replicating the mint to every node doesn't itself much add to the overall code complexity."

Correct.

"Adding barter note exchange markets, and automating the whole thing so that the mints and exchanges are largely invisible to the user, might. But AFAIK nothing like this has ever been tried."

It was attempted, but we never managed to get it to work well. In the end we punted and aimed for a simpler system that aimed to capture some of the key features of a true resource market (e.g. keeping the settlement currency fixed and centralized, replacing dynamic pricing with a simple Paris Metro pricing model based on "do it now" and "do it whenever" queues, etc.) You really have no idea what kind of a complexity barrier you are trying to overcome here. In MojoNation we were able to cheat by centralizing some bits and it was still a nightmare of complex interdependent components, each of which increased the fragility of the system.

"But it may also be brilliant to add to the under-the-covers complexity in order to increase the efficiency and flexibility of the overall system -- to make it more like a market."

As someone who has actually "been there, done that" I can tell you with absolute certainty that you are incorrect. Don't add complexity until you absolutely need it to solve an actual problem. Wait until you hit an efficiency or flexibility problem and then break out your toolkit of "clever" ideas, but until then keep them locked away lest you be tempted to actually use them.

10:12 PM

nick said...

"anonymous", I'd be quite interested in more detail about what you tried to implement and how it compares to what I have proposed. I read quite a bit of the MN documentation and don't recall seeing anything strongly resembling what I have proposed. I am not at all convinced that what you tried and what I have proposed are the same thing. There is a vast design space here, and I am extremely skeptical of any suggestion that your team explored more than a tiny fraction of it.

6:14 PM

Jim McCoy said...

"I'd be quite interested in more detail about what you tried to implement and how it compares to what I have proposed."

At some point we should sync up over vast quantities of alcohol and I can describe various approaches that were proposed, implemented, or abandoned.

"I read quite a bit of the MN documentation and don't recall seeing anything strongly resembling what I have proposed."

One of our many failure points was a very opaque design and development process. A lot of our various ideas and implementation changes were either lost when whiteboards were re-tasked (at one point we were actually using polaroid snapshots of whiteboards to maintain internal history :) or never explained even in the code. It was a rather frenetic time and no one really had the inclination to write up white papers or discuss design decisions in email, so we ended up leaving less of a digital trail than anyone would have liked. Oddly enough, the best description of the first couple of generations of the MN architecture did not even come from us, Bo Leuf actually dug through code to describe how things worked in his p2p book. By the time it was published we had already moved on to our paris metro pricing model, but it still captured a lot of the low-level details that were retasked for this third generation of the

system.

"I am not at all convinced that what you tried and what I have proposed are the same thing."

I think they are closer than you suspect, but other than offering a few warnings about building complicated solutions to solve self-imposed problems I will gladly step aside and give you the opportunity to make the same mistakes we did.

"There is a vast design space here, and I am extremely skeptical of any suggestion that your team explored more than a tiny fraction of it."

I agree, but I do know that we explored a lot of the same space you are considering here. We all come from the same set of influences and share the same goals, so I think you would be surprised at how much overlap there really was. There are most certainly a lot of details that can be tweaked and modified, but I would still maintain most of the core principles are still the same.

Perhaps you have some insight that we lacked, but I just want to offer the gentle suggestion that perhaps we did not miss any of the key points and in fact this design space resembles a desert than it does the promised land...

10:08 PM

nick said...

Jim, I do appreciate your comments, and I'd love to learn more over beer (but not until August -- bar exam :-).

I have extensively studied barter and money systems (in addition to working for David Chaum and playing a major role setting up a certificate authority) and do believe I have some insights.

Chief among these I recommend extensive thinking about mental transaction costs and proxy measures. (See links in original post or Google any of these phrases with my name to get my articles). A gigabyte is a gigabyte, Paris Metro pricing, and quality-of-service metrics are examples of proxy measures -- there are again a very large number of possibilities.

Along with many ways to measure a given resource, there are a wide variety of other Internet resources that might be commoditized: screen real estate, ad keywords, response time, CPU cycles, and much more. Setting up a general nanobarter architecture would be an interesting open source project in its own right -- it need not be just an expensive adjunct to a particular application. The number of ways this architecture could be used are vast and the best applications are probably not the most obvious ones.

As for complexity, I find it interesting that Donutlab was implemented in just 72 hour hours. One can get prototypes running rather quickly and then experiment with a wide variety of proxy measures.

As for the design space, it's a good bet that just as with traditional markets, some parts are desert, some parts are fertile, and some parts are positively jungle. But with nanomarkets we haven't yet explored any of this planet beyond site of the first lander. If there are any theoretical reasons to believe it's mostly desert -- beyond my warnings about the crucial role of user preferences and mental transaction costs -- I'd love to hear them.

3:55 PM

Jim McCoy said...

Nick,

I am absolutely certain that you are going to bring insights and ideas to the problem that we never considered back in the day, I just wanted to warn you that some of what you have discussed in this post and in the past are in fact things that were tried and abandoned in our short experiment of the interaction between these ideas and the real world. We too had former Digicash employees on staff, David Chaum and Andrew Odlyzko giving us free advice, etc. The hard part is not necessarily having the idea or insight, it is

in deciding when the application of same is worth the effort.

You are correct in stating that there are a lot of resources which can potentially be turned into commodities, but building a market for the resources does not mean that anyone will show up to play. This is the hard part, and you are competing against entrenched players whose centralized systems are large and ruthlessly efficient given the constraints of centralization. Getting a hook into something that people want to do where a resource market has enough breathing space to not get crushed by someone with a big, dumb, and simple alternative is a difficult hurdle to get over. As a simple example, the distributed storage model that something like amd-tahoe exemplifies is one of those areas that at first glance seems ideal for this sort of an application, but in fact is easier and cheaper to provide in a centralized manner. You need to find something that is either poorly served by centralization or where there are inherent inefficiencies in the centralized model that cannot be overcome by cheaper servers/bandwidth.

"As for complexity, I find it interesting that Donutlab was implemented in just 72 hour hours. One can get prototypes running rather quickly and then experiment with a wide variety of proxy measures."

I don't really find this an interesting example. The reality is that DonutLab was implemented in 72 hours + twelve years. 72 hours is what it took for the creators of a language that had a strong capability-security model to create a system which leveraged all of the advantages of this system without providing anything of value to someone whose code is not written in E, and twelve years is what it took to get E from Doug Barnes' weird hack of Java to a trustworthy system. Since you want people to actually use your system you are unlikely to be writing it in E, which means that you will need to reinvent a lot of wheels to get to the point where you can inject your system into an existing platform over a long weekend.

Complexity has a cost, and distributed systems are always more complex than centralized ones. All I would suggest is that before embarking on a resource-market quest is that you step back and apply your keen eye towards market efficiencies to the question of at what point a distributed, decentralized system will overtake a centralized competitor (if ever) and whether or not it seems realistic to assume you will reach this cross-over point.

11:10 AM

Anonymous said...

Hey, landed on your blog, nice stuff. I found a cool new tool for our blogs... www.widgetmate.com It helps get latest news for our keywords directly on to our blog. I added it on mine. Worked like a charm.

6:51 AM

Zooko said...

Nick:

I have two questions.

1. Why is nano-barter better than a fixed algorithm for controlling the automated exchange of small values, such as BitTorrent's Tit-For-Tat policy? (http://citeseer.ist.psu.edu/cohen03incentives.html) Or why is it better than a more sophisticated/general meta-algorithm such as those suggested in Computational Mechanism Design (http://www.eecs.harvard.edu/~parkes/mechdesign.html) ?

2. What should be the default initial buy/sell policy for nano-barter for storage-sharing such as in Allmydata Tahoe?

One possible answer to the first question is that nano-barter is in some sense equivalent to BitTorrent Tit-for-Tat and to Computational Mechanism Design, but that nano-barter explicitly recognizes that the scope of the market may expand in the future. That is: if you're thinking in terms of BT-TfT or Computational Mechanism Design, you're less likely to realize that people could if they chose start buying and selling the special-purpose currency units on eBay (just as people currently exchange virtual world game tokens on eBay).

The second question is a very hard practical question. One major failure of the agoric aspect of Mojo Nation was that we started by imagining that the default, or initial, policy implemented by the computational agent was not so important, since users would be incentived to tweak or replace that policy in order to profit. (This idea is sort of strangely similar to the Coase Theorem. :-))

In fact, we never achieved a real "market" in the sense of having heterogeneous strategies among the players -- more or less every Mojo Nation agent always used the strategy that it shipped with.

Thanks for your ideas. I look forward to more.

Regards,

Zooko

8:50 AM

nick said...

"1. Why is nano-barter better than a fixed algorithm for controlling the automated exchange of small values, such as BitTorrent's Tit-For-Tat policy?..."

Resource allocation algorithms and simple nonmarket barter schemes (e.g. tit-for-tat), although they often work well enough given the copious computational resources we have available, are usually quite suboptimal, because they make poor use of distributed knowledge and user preferences. See Hayek for example on how markets serve to communicate distributed and heterogenous knowledge.

That said, where user preferences and knowledge are homogenous and predictable, finding proxy measures (like Bit Torrent's "a megabyte is a megabyte" tit-for-tat barter) is more straightfoward and the case for markets over resource allocation or simple game algorithms is less compelling. Also, the surprisingly subtle nature of proxy measures and the mental transaction cost problem (which your description of Mojo's "tweaking" problem highlights) may often make it nontrivial to discover market mechanisms that work better than mature resource allocation algorithms. In other words, there may be quite a bit of "desert" to explore before one hits on an "oasis": good choices and combinations of user input, proxy measures, and markets that bring out the superiority of markets as resource allocators where user preferences and knowledge are heterogenous.

I greatly wish economists and computer scientists would join up and do research in this area, designing markets and comparing them head-to-head with resource allocation algorithms. That missing, the engineer will have to perform his own experiments.

Despite these caveats, the frequent superiority of markets in the traditional economy, as well as the theoretical proofs of its efficiency by economists and the heterogenous nature of user preferences on the Internet, all point to a vast potential for nanomarkets.

Z: "2. What should be the default initial buy/sell policy for nano-barter for storage-sharing such as in Allmydata Tahoe?"

I'm sure you and Jim can make better guesses than I in this regard, and I'd love to hear them, but I can suggest a general approach. My approach would be to start with some possible proxy measures of what users find valuable in backup services: gigabytes, reliability, timeliness, etc. Choice(s) of proxy measure are worth quite a bit of brainstorming and experimentation because the best proxy measures may be highly non-obvious. (Think of the paradoxical nature of using time as a proxy measure for labor value, for example, yet that has commonly proved to be a superior measure compared to, for example, the number of widgets produced). Then make some educated guesses about how variable these user preferences are (some customers care more about raw storage capacity, some more about reliability, etc.) Figuring out what users want may also suggest new kinds of proxy measures you hadn't thought of.

The variability (heterogeneity) of preferences also suggests whether a simple algorithm or a more sophisticated market will work better, and if there is heterogeneity what proxy measures of value (gigabytes, timeliness, reliability, etc.) the market should price and trade.

Finally, buy/sell orders should be based on the variable user behavior. The engineer should use his knowledge of this behavior to create algorithms that translate it real-time into buy and sell orders for gigabytes, timeliness, reliability, etc. The more user preferences and knowledge can be captured without bothering the user (mental transaction costs), the better the market will work relative to simple schemes.

"One major failure of the agoric aspect of Mojo Nation was that we started by imagining that the default, or initial, policy implemented by the computational agent was not so important, since users would be incentived to tweak or replace that policy in order to profit. (This idea is sort of strangely similar to the Coase Theorem. :-))...In fact, we never achieved a real "market" in the sense of having heterogeneous strategies among the players -- more or less every Mojo Nation agent always used the strategy that it shipped with."

The problem here seems to be trying to get preferences directly from special user input for that purpose rather than indirectly from normal user behavior. (Furthermore, I remember trying to "tweak" Mojo myself and it was hardly a user friendly experience :-) As your Coase comment suggests, it is easy in the essential quest for user preferences for the mental transaction costs to smother the value added by the nanomarket.

4:40 PM

George Weinberg said...

Are there systems out there that make it easy for everyone who wants to to operate a mint? I'm not aware of any, I've spent some time working on one of my own but I really don't want to respec the wheel.

6:06 PM

nick said...

There used to be open source code out there that went by the name of "lucre" and "lucrative". I don't know how easy these made it to set up a mint.

12:45 PM

Zooko said...

One thing I really don't like about blogs is that conversations tend to have a "lifespan" of at most a couple of weeks. I'm very interested in this topic, and I have many unanswered questions, but my guess is that few or no people are ever going to even see this comment.

Anyway, I still am not sure I understand precisely what is the difference, if any, between the following three ways to allocate computational/storage/network resources:

* automated markets (agorics) e.g. Mojo Nation

* distributed game-theoretic mechanism design e.g. BitTorrent

* push the whole issue to a higher layer e.g. Amazon S3

* nano-barter

This is perhaps a subtler question than it first appears because the closer I look the blurrier appear the distinctions between these approaches. For example, suppose you want to use a fully agoric approach, so that all of our reasoning about markets (markets of humans) can be applies to the automated market. Well, this is basically supposing that -- at least as far as economics goes -- the computational agents that make up the market are indistinguishable from human beings! That's a pretty strong assumption. But then suppose you weaken it by saying that, while these agents are not fully rational, utility-optimising thinkers, they are at least capable of optimizing within a limited set of resources and strategies. That way lies game theoretic mechanism design. But again, of course the agent doesn't choose to act based on its own value system, instead it obeys the values programmed into it by a user. Then we're halfway to a modern resource such as Amazon S3 or the way many ISPs work, where the humans agree on prices, limits and a general contract, and then automated mechanisms simply measure aggregate usage and settle up.

So, given that I don't have a solid grasp on this issue, my current strategy with Allmydata Tahoe is to delay committing to any of these strategies. What is necessary to enable any of these strategies later? I guess it is (a) that the use of resource in question can be measured and (b) that it can be programmatically allowed/denied.

2:31 PM

nick said...

Z: "One thing I really don't like about blogs is that conversations tend to have a "lifespan" of at most a couple of weeks. I'm very interested in this topic, and I have many unanswered questions, but my guess is that few or no people are ever going to even see this comment."

The short lifespan for me comes because it's too much work to keep going back to all my old posts to see if there are new comments. Luckily for us I had some reason to come back to this one and caught yours here. Now lets see if you come back and read this. :-)

Z: "I still am not sure I understand precisely what is the difference, if any, between the following three ways to allocate computational/storage/network resources:

* automated markets (agorics) e.g. Mojo Nation

* distributed game-theoretic mechanism design e.g. BitTorrent

* push the whole issue to a higher layer e.g. Amazon S3

* nano-barter"

Nano-barter is mostly just my own version of agorics that emphasizes the necessity of obtaining user preferences in low mental transaction ways. Once one has obtained good information on user preferences the lower limit on granularity is extremely small (thus "nano" to distinguish from mere "micropayments", or we also might call it "kiloherz commerce" or "megaherz commerce" at some point).

"Well, this is basically supposing that -- at least as far as economics goes -- the computational agents that make up the market are indistinguishable from human beings!"

Even if computational agents could be intelligent like humans, which they are nowhere close to being, they still couldn't read their masters' minds. They have to tediously observe their master to get any useful information. This is why a fully automated market that ignores user behaviors won't work. It will be useless. A market or game mechanism that works has to be based on informed rather than speculative guesses by the software about what users actually want during this next microsecond.

"they are at least capable of optimizing within a limited set of resources and strategies."

Not even this is true unless the agents have good information about their masters' behaviors from which can at least crudely be guessed important unique aspects of their masters' preferences.

"What is necessary to enable any of these strategies later? I guess it is (a) that the use of resource in question can be measured and (b) that it can be programmatically allowed/denied."

I'd say you need more to experiment with several approaches than to delay decision. And the kind of measurement should also be something you experiment with -- it would be foolish to commit to, say, charging by the gigabyte (rather than by reliability, alacrity, or combinations of these or other factors) early. In fact, an architecture that allows deploying new game mechanisms or markets based on new proxy measures would be quite useful.

2:22 PM

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Simple theme. Powered by Blogger.